Deep Dynamic Mixed Membership Stochastic Blockmodel

Zheng Yu, Marcin Pietrasik, Marek Reformat

Problem Formulation

- Given a network of entities and the relationships between them, our goal is to model the network by learning the underlying distributions that govern the entity relations.
    - We can treat the input network as a graph, represented by a binary adjacency matrix, $\mathbf{X}$, such that entry $\mathbf{X}_{ij}$ equals 1 if there is a relation from entity i to entity j and is 0 otherwise.
- We can evaluate the quality of our model by masking certain relations in the network and seeing whether the model can successfully predict the unobserved relations.

Mixed Membership Stochastic Blockmodel Approach to Network Modeling

- The Mixed Membership Stochastic Blockmodel (MMSB) decomposes the network into communities that share similar properties and assigns entities with membership to them.
- Community relations are modeled as the degree of similarity between communities in the community relations matrix.
- Relations between two entities are modeled as the community relations of their respective communities.
- The building blocks (entity memberships, community relations) are inferred using Bayesian inference approximation algorithms.

Mixed Membership Stochastic Blockmodel Generative Model

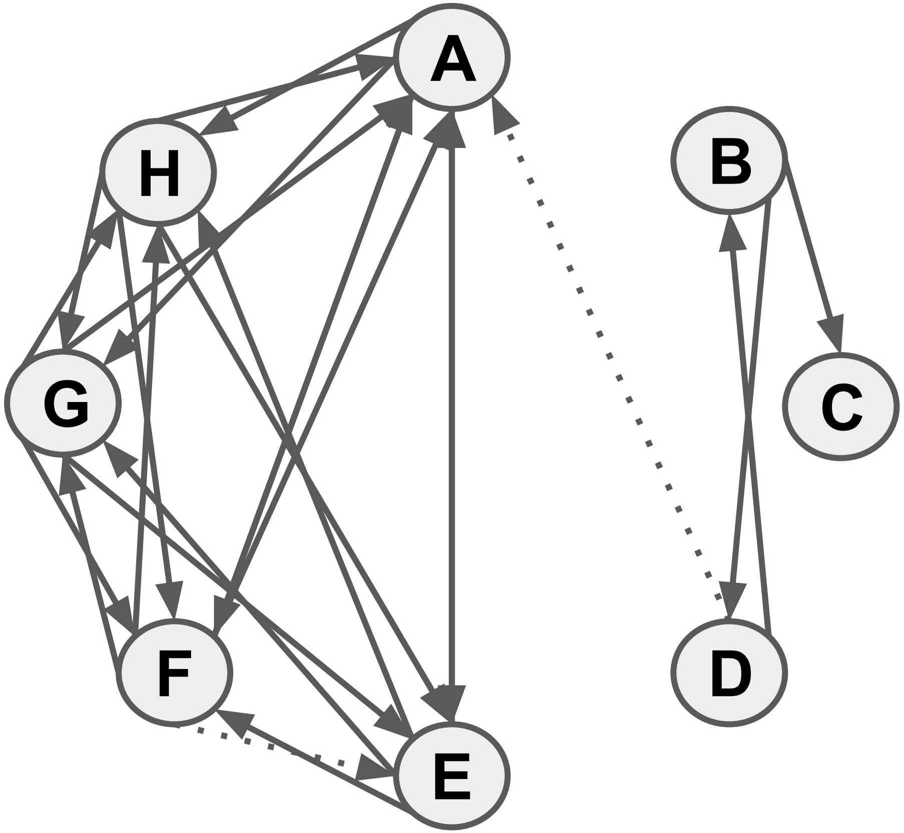
- For each entity i
  - Draw membership distribution $\theta_i \sim \text{Dirichlet}(\alpha)$
- For each entry pq in community relations matrix **B**
  - Draw community relation $b_{pq} \sim \text{Beta}(\lambda_1, \lambda_2)$
- For each relation from entity i to entity j
  - Draw sender's indicator $\mathbf{z}_{i\rightarrow j} \sim \text{Multi}(\theta_i)$
  - Draw receiver's indicator $\mathbf{z}_{i\leftarrow j} \sim \text{Multi}(\theta_j)$
  - Draw relation $x_{ij} \sim \text{Bernoulli}(\mathbf{z}_{i\rightarrow j}\mathbf{B}\mathbf{z}_{i\leftarrow j})$

entity

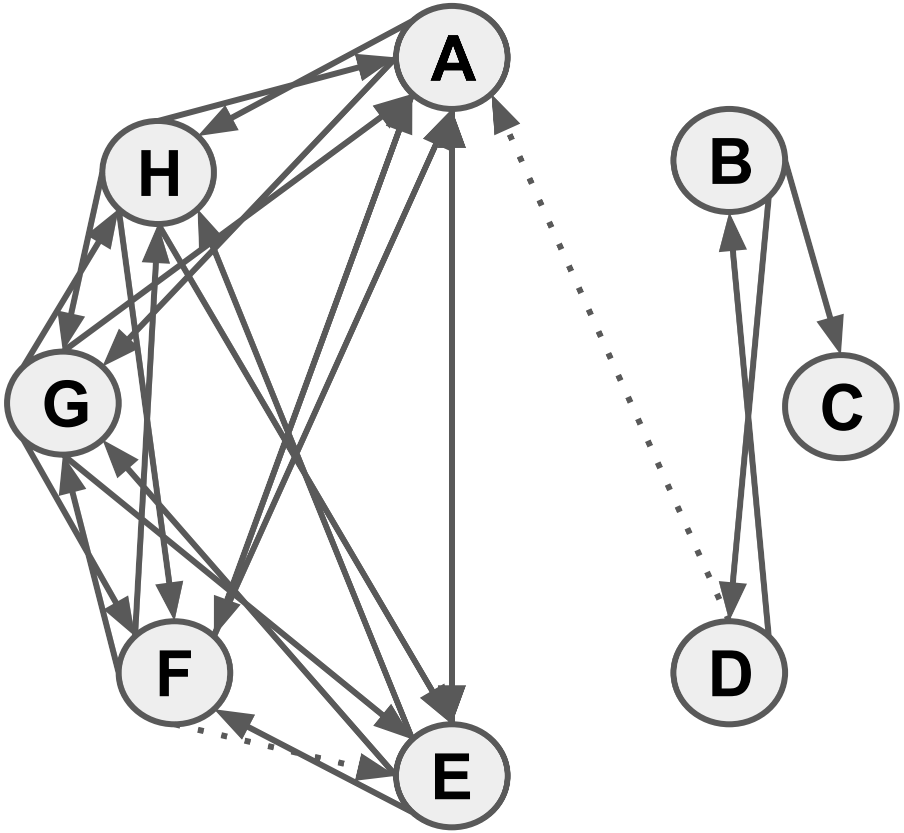
relation

unknown

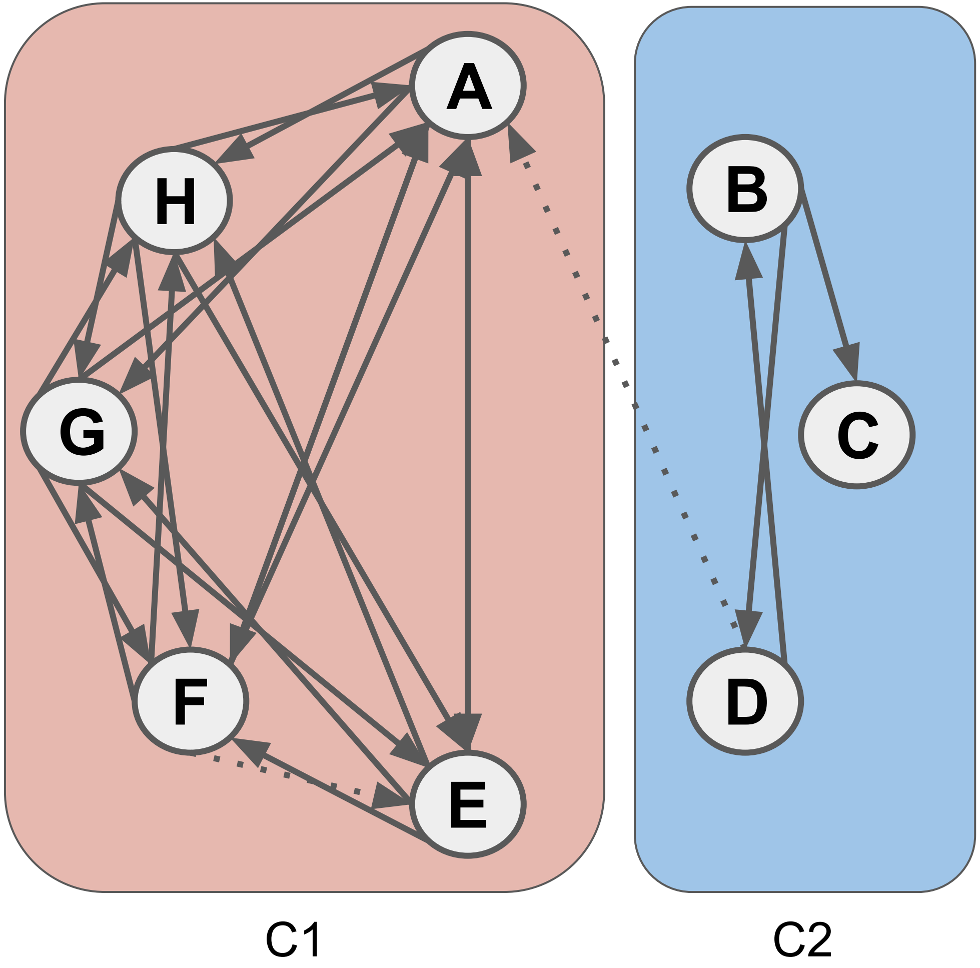
no relation



0	0	0	0	1	1	1	1
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0
?	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1
1	0	0	0	?	0	1	1
1	0	0	0	1	1	0	1
1	0	0	0	1	1	1	0

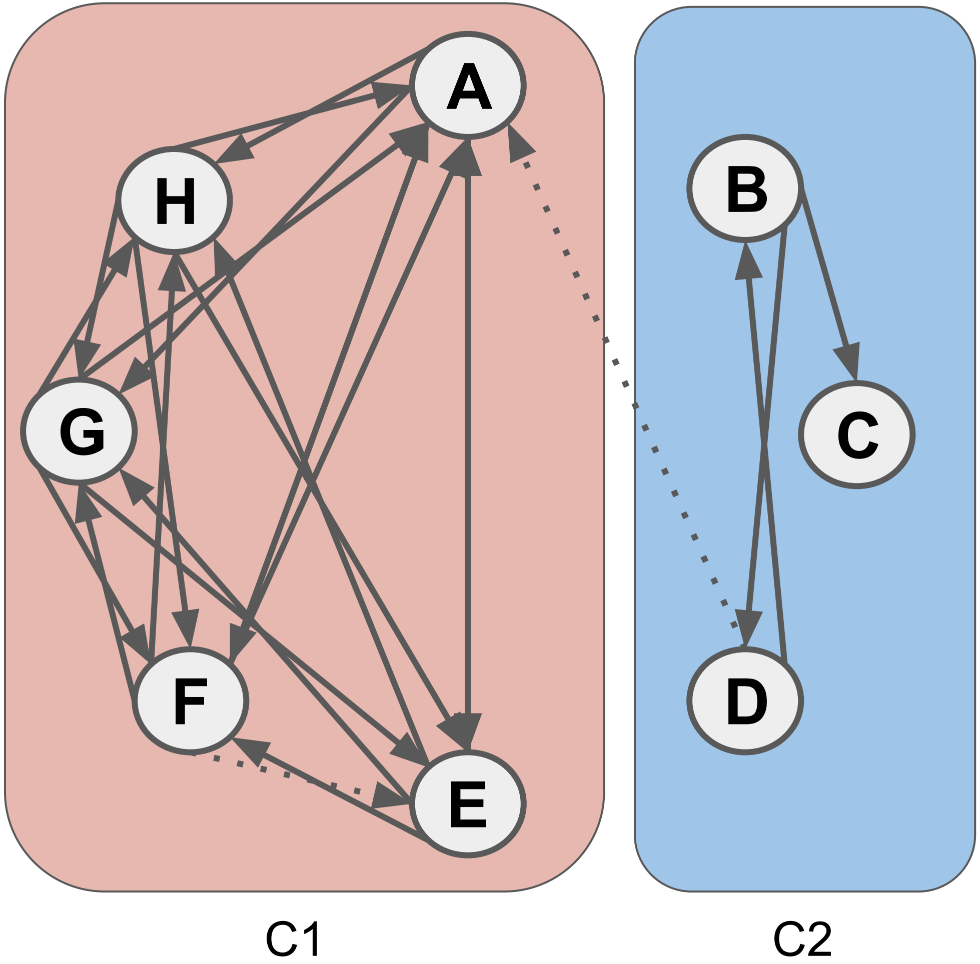


- Assign each entity to a community. (Assume $\mathbf{z}_{i \rightarrow j} = \mathbf{z}_{i \leftarrow j}$)



- Assign each entity to a community. (Assume $\mathbf{z}_{i\rightarrow j} = \mathbf{z}_{i\leftarrow j}$)

	C1	C2
C1	1	0
C2	0	½

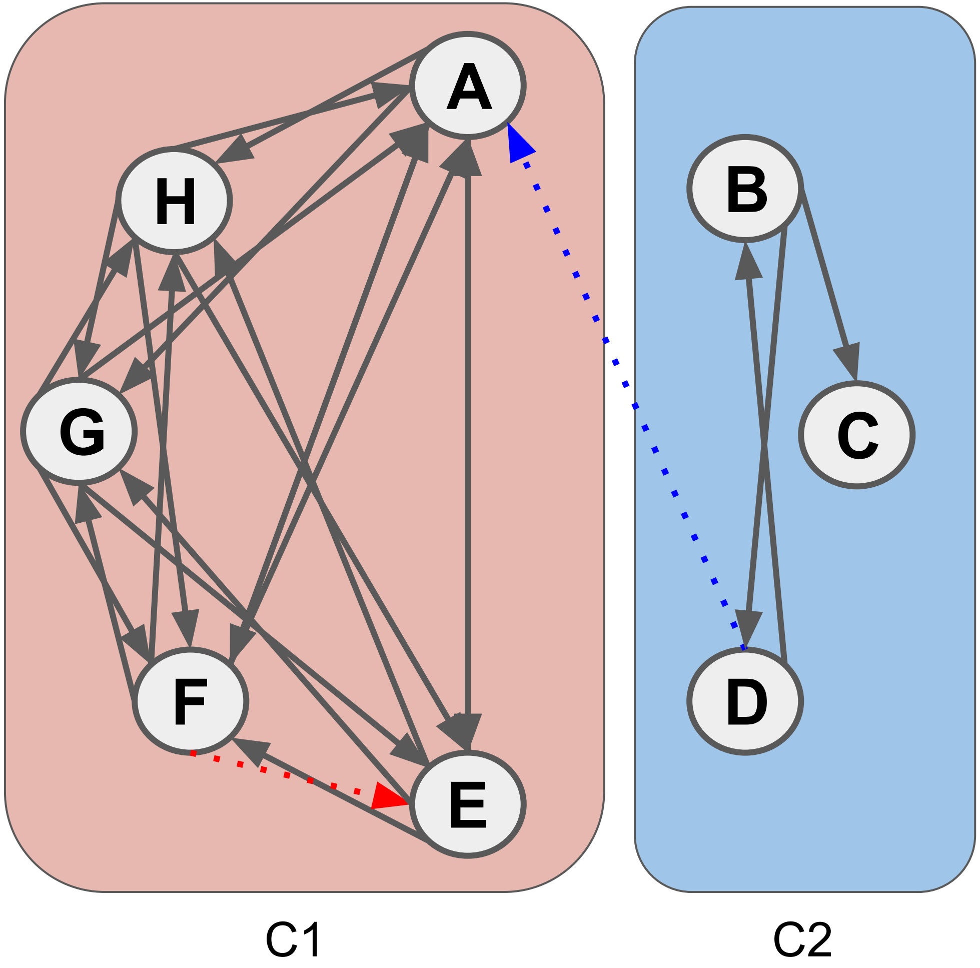


- Assign each entity to a community. (Assume $\mathbf{z}_{i\rightarrow j} = \mathbf{z}_{i\leftarrow j}$)

	C1	C2
C1	1	0
C2	0	½

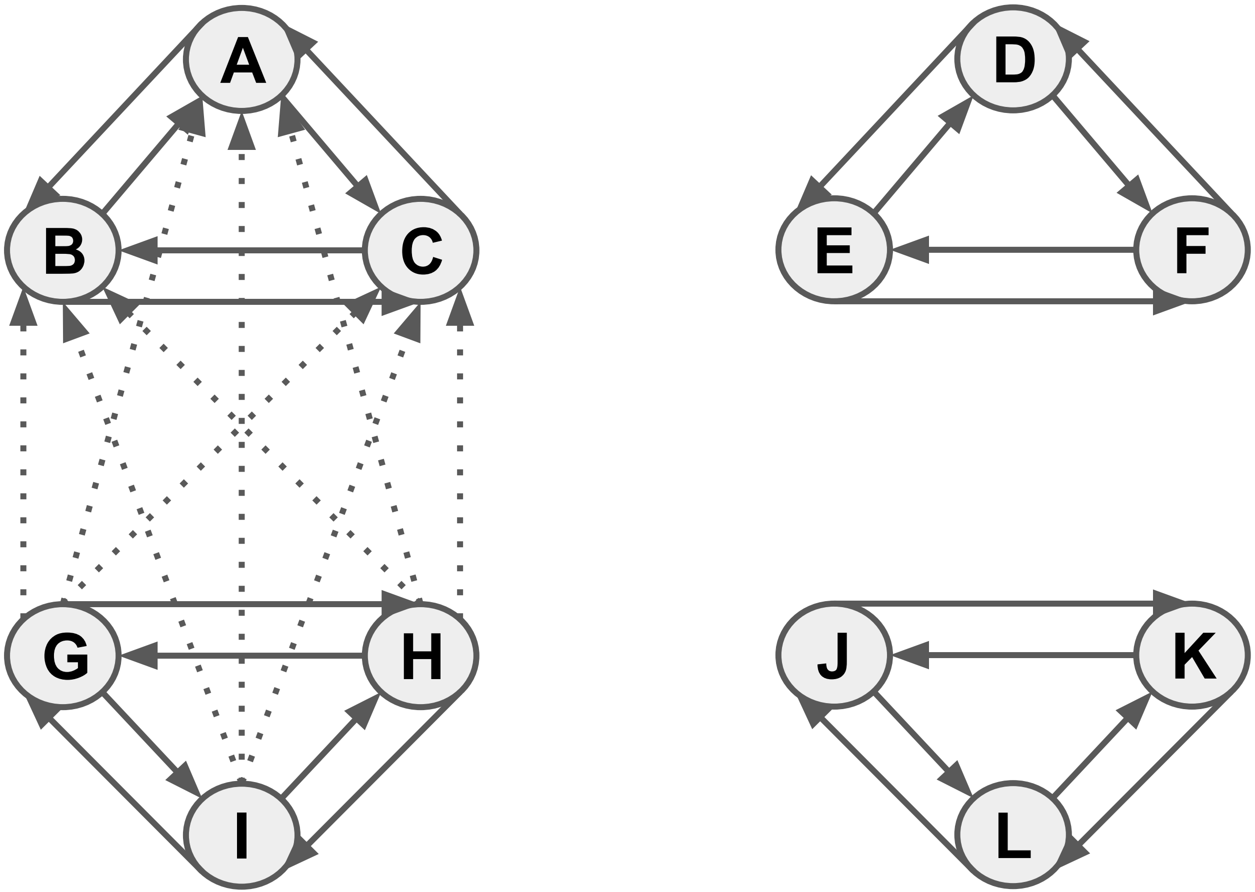
P(F → E) ~ Bernoulli(1)

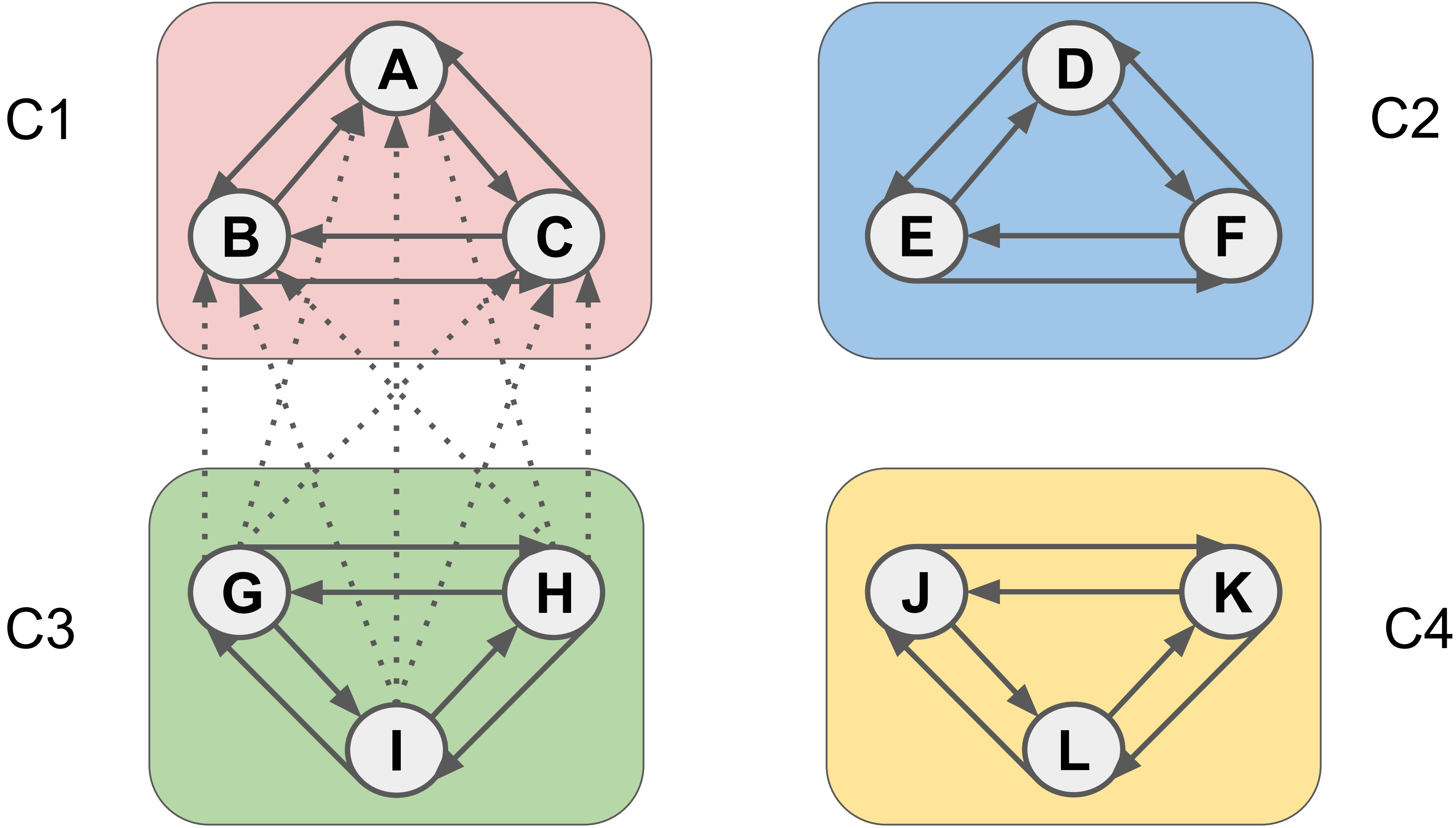
P(D → A) ~ Bernoulli(0)



Problem with MMSB

- When all entity relations between two communities are unknown, the community relation is reduced to its prior.
- Recall that b_{pq} ~ Beta(λ_1, λ_2) and **B** is inferred on its posterior.
    - $p(\mathbf{B} \mid \mathbf{X}, \lambda_1, \lambda_2) \sim p(\mathbf{X} \mid \mathbf{B})\, p(\mathbf{B} \mid \lambda_1, \lambda_2)$
- If all entity relations are unknown, the posterior of **B** is reduced to its prior.
    - $p(\mathbf{B} \mid \lambda_1, \lambda_2)$
- If all entity relations from community *p* to community *q* are unknown, the posterior of b_{pq} is reduced to its prior.
    - $p(b_{pq} \mid \lambda_1, \lambda_2)$

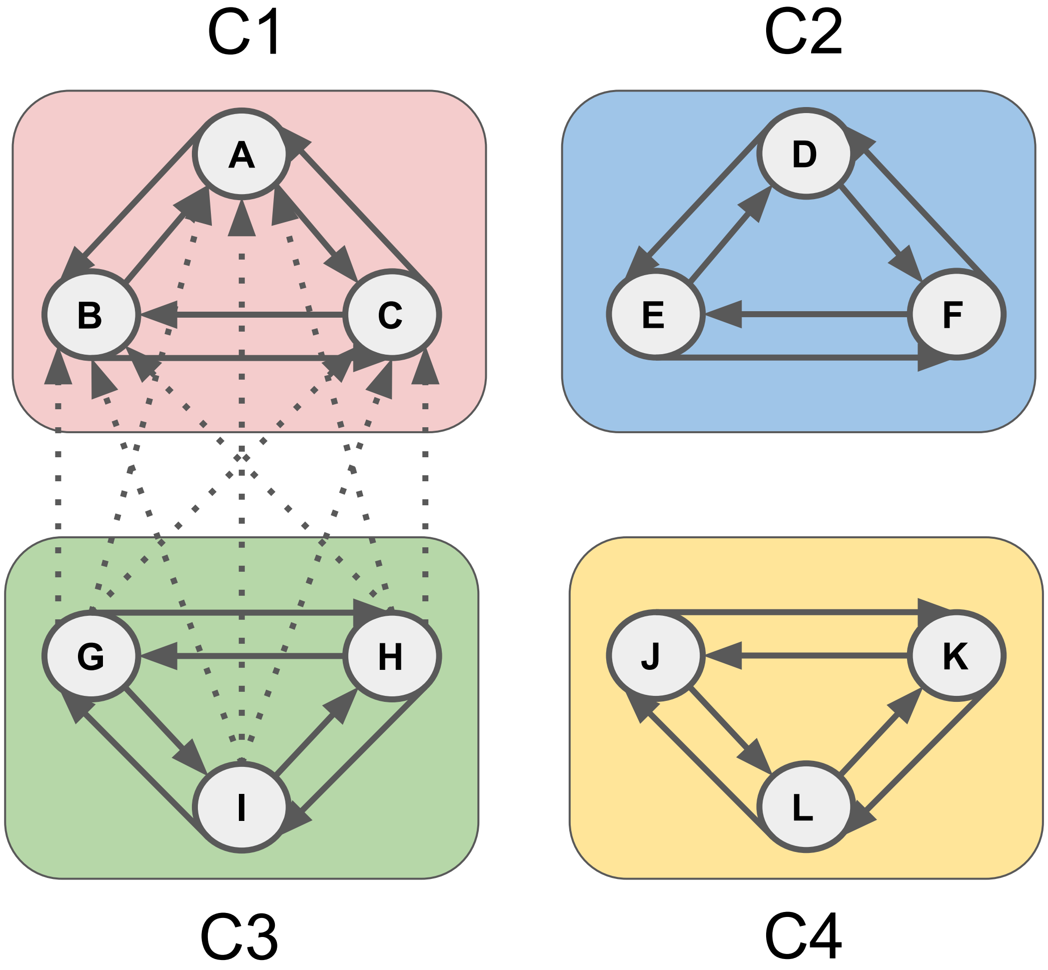

A
B
C
D
E
F
G
H
I
J
K
L



C1
A
B
C
C2
D
E
F
C3
G
H
I
C4
J
K
L

	C1	C2	C3	C4
C1	1	0	0	0
C2	0	1	0	0
C3	?	0	1	0
C4	0	0	0	1

No way to infer relation from community C3 to community C1.

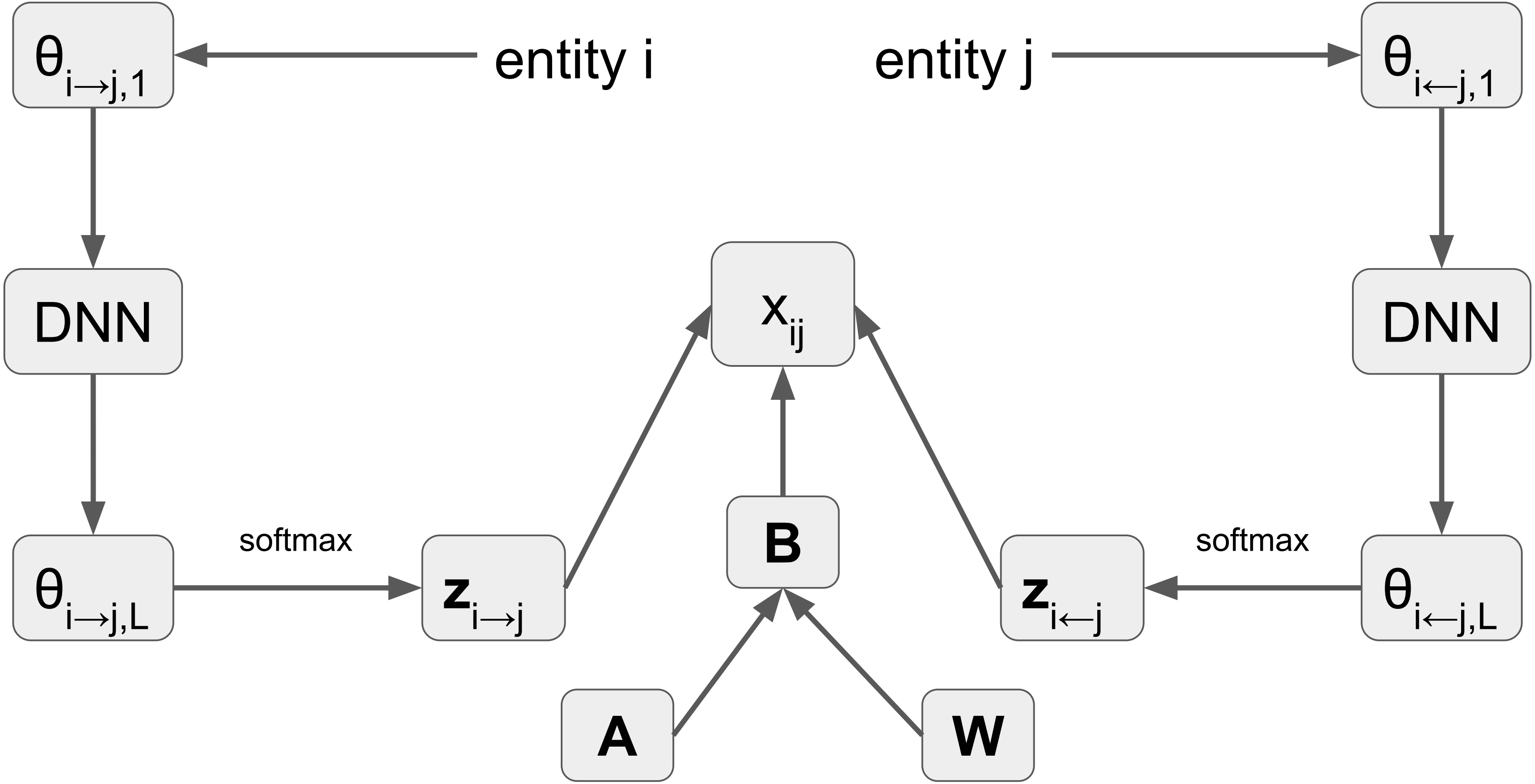


Our Solution

- Factorize community relations matrix, **B**, into two community feature matrices **A** and **W**.
  - The relation b_{pq} is modeled as the dot product between $\mathbf{a}_p$ and $\mathbf{w}^T_{\mathbf{q}}$
  - This introduces a dependency between communities, allowing our model to draw information from observed community relations to infer unobserved community relations.
- Replace the probabilistic framework of the MMSB with a deep neural network (DNN).
- Introduce a temporal dependence via a long-short term memory recurrent neural network (LSTM).

Static Generative Model

- For each relation from entity i to entity j
  - For each layer l in deep neural network
    - Update $\boldsymbol{\theta}_{i\rightarrow j,l} = \sigma(\mathbf{R}_{S,l}\boldsymbol{\theta}_{i\rightarrow j,l-1} + \mathbf{d}_{S,l})$
    - Update $\boldsymbol{\theta}_{i\leftarrow j,l} = \sigma(\mathbf{R}_{R,l}\boldsymbol{\theta}_{i\leftarrow j,l-1} + \mathbf{d}_{R,l})$
  - Update sender's membership indicator $\mathbf{z}_{i\rightarrow j} = \mathrm{Softmax}(\mathbf{R}_{S,z}\boldsymbol{\theta}_{i\rightarrow j,L} + \mathbf{d}_{S,z})$
  - Update receiver's membership indicator $\mathbf{z}_{i\leftarrow j} = \mathrm{Softmax}(\mathbf{R}_{R,z}\boldsymbol{\theta}_{i\leftarrow j,L} + \mathbf{d}_{R,z})$
  - Draw relation $x_{ij} \sim \mathrm{Bernoulli}(\sigma(\mathbf{z}_{i\rightarrow j}[\mathbf{AW}]\mathbf{z}_{i\leftarrow j}))$


θ
i→j,1
entity i
entity j
θ
i←j,1
DNN
DNN
x
ij
θ
i→j,L
softmax
z
i→j
B
z
i←j
softmax
θ
i←j,L
A
W

Dynamic Model

- For dynamic networks, we use a long-short term memory (LSTM) component to model the temporal changes in the network.
  - LSTM predicts latent entity features, $\boldsymbol{\theta}^t_{i\rightarrow j,l}$ and $\boldsymbol{\theta}^t_{i\leftarrow j,l}$,from the community membership indicators from the previous time step, $\mathbf{z}^{t-1}_{i\rightarrow j}$ and $\mathbf{z}^{t-1}_{i\leftarrow j}$.
- Community relations matrix, **B**, and network weights, **R** and **d**, are shared across time steps.

Evaluation

- We evaluate the static and dynamic models on the link prediction task separately, using five real-world datasets:
  - Static: NIPS, MIT, Lazega.
  - Dynamic: Coleman, Email.
- Each dataset is split into training (80%) and testing (20%) subsets. The models are trained on the training subsets five times and evaluated based on the predicted links on the testing subsets.
- The metric for comparison is the area under the receiver operating characteristic curve (AUC).

Static Results

Model	NIPS	MIT	Lazega
IRM	0.8901 ± 0.0162	0.8261 ± 0.0047	0.7056 ± 0.0167
LFRM	0.9348 ± 0.1667	0.8529 ± 0.0179	0.8170 ± 0.0197
MMSB	0.9524 ± 0.0215	0.8561 ± 0.0176	0.7989 ± 0.0102
iMMM	0.9574 ± 0.0155	0.8617 ± 0.0124	0.8074 ± 0.0141
NMDR	—	0.8569 ± 0.0138	0.8285 ± 0.0114
cMMSB	0.9581 ± 0.0153	0.8794 ± 0.0159	0.8273 ± 0.0148
DDBN	**0.9660 ± 0.0064**	**0.9040 ± 0.0055**	**0.8550 ± 0.0054**

Dynamic Results

Model	Coleman	Email
Common Neighbour	0.8794 ± 0.210	0.9120 ± 0.0029
Jaccard Coefficient	0.8821 ± 0.0196	0.9057 ± 0.0027
Adamic/Adar	0.8823 ± 0.0204	0.9186 ± 0.0029
MNE	0.8990 ± 0.203	0.8816 ± 0.0045
DeepWalk	**0.9107 ± 0.0221**	0.7605 ± 0.0051
PMNE	0.9085 ± 0.0119	0.7598 ± 0.0062
BPTF	0.8895 ± 0.0246	**0.9592 ± 0.0149**
DDBN	0.8920 ± 0.0067	0.9481 ± 0.0026

Summary

- Described the problem of missing community relations in network modeling.
- Introduced the DDBN which solves this problem by factoring community relations matrix **B** into feature matrices **A** and **W**.
- Our model achieves a performance that is comparable to or better than state-of-the-art methods on the link prediction task.

Questions?